

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
Факультет прикладної математики та інформатики
Кафедра прикладної математики

Дипломна робота

РОЗРОБКА WEB - МАГАЗИНУ ТА ЙОГО РЕАЛІЗАЦІЯ З
ВИКОРИСТАННЯМ FREEWARE

Виконала: студентка групи ПМП-42
спеціальності
113 - прикладна математика
Дулиба Д. М.

_____ (прізвище та ініціали)

Керівник Дияк І. І
_____ (прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

ЗМІСТ

ВСТУП.....	2
РОЗДІЛ 1. РЕАЛІЗАЦІЯ ПРОЄКТУ ТА АКТУАЛЬНІСТЬ РОБОТИ.....	5
1.1. Реалізація проєкту.....	5
1.2. Актуальність роботи.....	6
РОЗДІЛ 2. ВИКОРИСТАНІ ТЕХНОЛОГІЇ.....	8
2.1. React.js.....	8
2.2. Node.js.....	9
2.3. Express JS.....	10
2.4. CORS.....	11
РОЗДІЛ 3. ОПИС ПРОЄКТУ.....	14
3.1. Telegram-бот.....	14
3.2. Робота з Firebase.....	16
3.3. UI частина інтернет-магазину.....	22
3.4. Функціонал інтернет-магазину.....	23
РОЗДІЛ 4. ХОСТИНГ ІНТЕРНЕТ-МАГАЗИНУ.....	31
ВИСНОВОК.....	32
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	34

ВСТУП

Інтернет-магазин - це веб-сайт з каталогом товарів, який дозволяє клієнтам купувати і доставляти товари безпосередньо на сайті. Він відрізняється від звичайного магазину тим, що всі ці функції виконує програмне забезпечення, тому немає потреби у фізичних приміщеннях, торгових залах, вітринах, продавцях чи консультантах.

Веб-сторінки магазинів - це перспективний спосіб розвитку вашого бізнесу, особливо якщо у вас широкий асортимент товарів і ви часто його оновлюєте. Перевагами є те, що їх можна створити швидко і дешево, товари можна постачати з будь-якої точки світу, вони доступні 24/7 і мають низьку собівартість. Недоліки полягають у тому, що товари не можна повністю перевірити, але завжди є можливість повернути товар протягом 14 днів або отримати доступ нього перед оплатою.

Станом на початок 2023 року у світі налічується 5,16 мільярд користувачів Інтернету. Це означає, що 64,4% населення світу мають доступ до інтернету. За рік кількість користувачів Інтернету збільшилася на 1,9%. Кількість українських інтернет-користувачів також зросла і становить понад 32 мільйони, що складає приблизно 72% населення країни.

Онлайн-шопінг - дуже цікавий приклад зміни цифрових звичок. Існує багато свідчень того, що зараз люди є більш активними користувачами електронної комерції, ніж до пандемії.

До пандемії онлайн-покупці купували товари в таких категоріях, як мода, краса та електроніка. Дуже мало людей замовляли їжу та напої. Більшість людей навіть не думали про те, щоб купувати їжу або товари для дому онлайн. Але зараз ситуація змінилася. Як тільки люди навчилися це робити, вони відчули зручність і гнучкість онлайн-покупок і швидко звикли до них.

Але ця тенденція має більше значення, ніж здається на перший погляд. Дані показують, що як тільки антиковідні правила були скасовані, люди з радістю повернулися до фізичних магазинів.

Водночас дослідження Statista показує, що, незважаючи на загальне зниження світових витрат на роздрібну торгівлю (онлайн і офлайн) минулого року, частка онлайн-витрат у 2022 році вища, ніж у 2021 році.

Згідно з тим же дослідженням, минулого року лише 17,1% світових роздрібних витрат було здійснено онлайн. Це означає, що лише шоста частина світових споживчих витрат здійснюється через електронну комерцію.

Більше того, тепер, коли багато людей мають свободу вибирати, де їм робити покупки, вони, безсумнівно, почнуть поєднувати онлайн і офлайн шопінг.

Важливо враховувати, що сьогодні більше людей обізнані про можливості електронної комерції, ніж до пандемії, і вони набагато частіше, ніж тоді, активно користуються сайтами для онлайн-шопінгу. Цю гіпотезу також підтверджують дані. Згідно з останніми прогнозами GWI та Statista, електронна комерція продовжить збільшувати свою частку в найближчі роки.

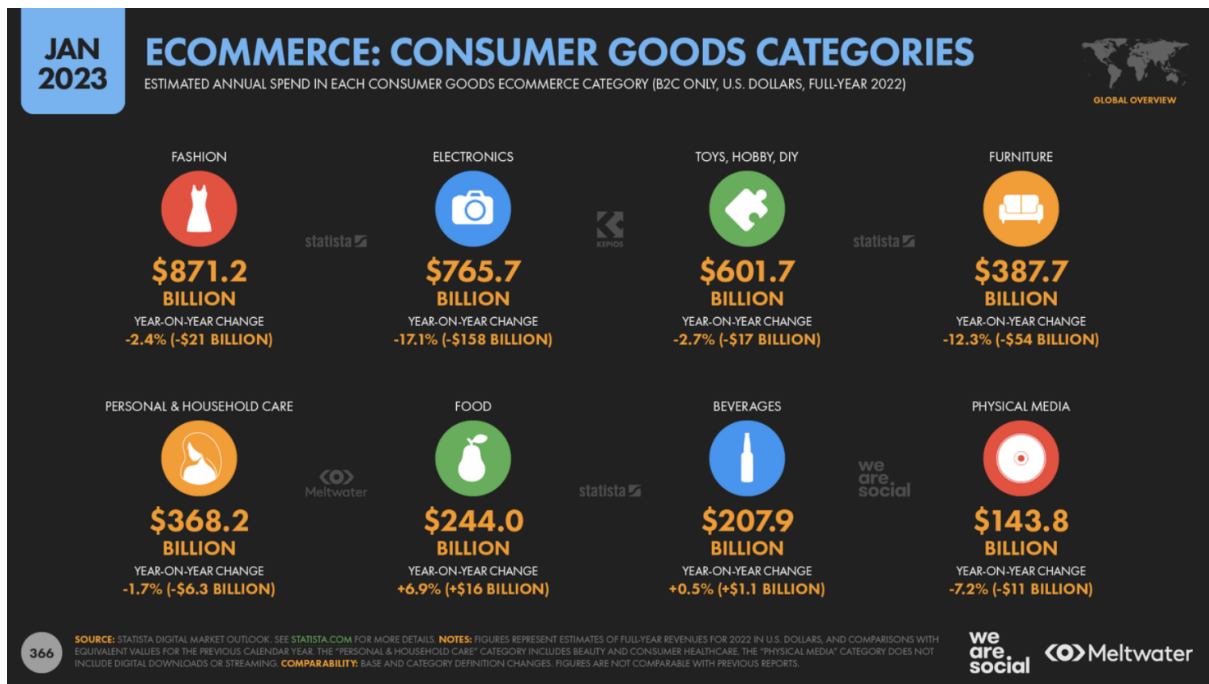


Рис. 1 Категорії споживчих товарів в Інтернеті

Загалом, зростання кількості користувачів Інтернету та попиту на електронну комерцію сприяє розвитку цифрової економіки України та входженню України у світовий інформаційний простір.

Тому розвиток інтернет-магазинів, здатних генерувати високі доходи при мінімальних витратах, має велике значення в секторі електронної комерції.

РОЗДІЛ 1. РЕАЛІЗАЦІЯ ПРОЄКТУ ТА АКТУАЛЬНІСТЬ РОБОТИ

1.1. Реалізація проєкту

Цей розділ присвячено реалізації проєкту інтернет-магазину на платформі React.js з використанням бази даних Firebase. Особливістю проєкту є надзвичайна швидкість, з якою власник може оперативно отримувати інформацію через спеціально розробленого Telegram-бота. Бот дає змогу власникам отримувати сповіщення про дані клієнтів і замовлення одразу ж у свій акаунт Telegram.

Крім того, у даній роботі було впроваджено базу даних Firebase для забезпечення надійного зберігання та обробки інформації в каталозі товарів. Firebase дає змогу забезпечити стабільну та безперебійну роботу магазину та зручне управління даними.

Для забезпечення взаємодії з ботом Telegram у проєкті використовується бібліотека telegram-bot-api. Це дає змогу взаємодіяти з ботом через зручний та інтуїтивно зрозумілий інтерфейс, спрощуючи його використання та забезпечуючи зручний доступ до інформації про замовлення.

Даний проєкт містить такі сторінки:

- Сторінка - каталог
- Сторінка “Про нас”
- Сторінка “Контакти”
- Сторінка “Ваша корзина”
- Сторінка “Оформлення замовлення”
- Сторінка “Відгуки”

Сторінка-каталог - це і є головна сторінка, на якій розміщені товари зі змогою сортувати їх за категоріями. Товари мають таку інформацію, як фото, назва, опис, ціна. Сторінка “Про нас” містить короткий опис про

магазин та карусель фото. На сторінці “Контакти” розміщені контакти магазину, на які після кліку можна зразу зв'язатися, та карта з посиланням на Google карти. Для можливості залишити відгук, створена сторінка “Відгуки”, яка має зручний інтерфейс. Сторінка “Ваша корзина” зберігає інформацію про ваше замовлення і підраховує загальну суму. На цій сторінці є можливість редагувати замовлення, а саме збільшити чи зменшити кількість або ж повністю видалити замовлення. Після натиску на кнопку “оформити замовлення” нас посилає на сторінку оформлення замовлення, яка містить поля для вводу даних. Після заповнення і підтвердження ці дані присилаються мені в telegram бот.

1.2. Актуальність роботи

Онлайн-бізнес залишається одним із найперспективніших секторів, оскільки його можна створити, маючи відносно невеликі ресурси порівняно з офлайн-кав'ярнями та магазинами.

Електронна комерція сильно змінила ринок торгівлі. Сьогодні для покупців відкриті всі можливі способи покупки, які не мають жодних обмежень. Вони можуть зробити замовлення з будь-якої точки світу та отримати його з кур'єром до свого порогу. Відкриття кордонів, широке використання платежів за замовлення з інших країн та стрімкий розвиток логістичних компаній створюють можливості як для компаній, що розширюють свої ринки та збільшують продажі, так і для споживачів, які мають необмежений вибір товарів та послуг.

Більшість населення України надає перевагу онлайн-покупкам, оскільки це заощаджує час. За даними дослідження Statista, витрати в онлайн ресурсах в 2022 році більші, ніж були у 2021. Отже, з кожним роком попит на електронну комерцію буде лише збільшуватися.

Актуальність інтернет-магазинів квітів у сучасних тенденціях електронної комерції безперечна. Квіти завжди були популярним подарунком, символом любові, вираженням вдячності та уваги. З цієї причини інтернет-магазини квітів пропонують широкий вибір свіжих квітів і рослин, які надійно доставляють додому покупцеві.

Основною перевагою інтернет-магазину квітів є зручність і швидкість покупки. Клієнти можуть замовити букети та квіткові композиції в будь-яку пору року, та доставку прямо додому або в офіс. Ця послуга особливо зручна для тих, у кого мало часу, або у віддалених районах, де мало можливостей купити свіжі квіти.

Сервіс також дозволяє клієнтам надсилати свої привітання разом з букетом. Це особливо корисно у випадках, коли клієнт не може бути присутнім особисто, але хоче передати свої почуття іншим.

Сьогодні електронна комерція стрімко зростає, а клієнти все більше звикають замовляти товари та послуги через Інтернет. Ця тенденція особливо помітна в Україні, де розвиток онлайн-торгівлі став пріоритетним напрямком розвитку бізнесу.

Тому інтернет-магазини мають великий потенціал і значення в сучасному суспільстві, де клієнти цінують зручність, широкий асортимент товарів та якість обслуговування. Вони можуть задовольнити потреби своїх клієнтів, доставляючи свіжі квіти і чудові букети швидко і надійно до їхнього порогу.

РОЗДІЛ 2. ВИКОРИСТАНІ ТЕХНОЛОГІЇ

a2.1. React.js.

Фреймворк React.js - це відкритий фреймворк та бібліотека JavaScript, розроблені компанією Facebook. React.js використовується для швидкої та ефективної розробки інтерактивних інтерфейсів користувача та веб-додатків з використанням значно меншої кількості коду порівняно з чистим JavaScript.

У React додатки розробляються шляхом створення багаторазових компонентів. Ці компоненти є окремими частинами інтерфейсу кінцевого користувача, які в поєднанні формують весь користувацький інтерфейс.

Головна роль React.js у додатку - це обробка рівня відображення (view) так, як у шаблоні model-view-controller (MVC), що гарантує ефективне виконання реалізації. Замість роботи з усім інтерфейсом користувача як єдиним цілим, React.js спонукає девелоперів ділити інтерфейси на окремі перевикористовувані компоненти, що утворюють будівельні блоки всього інтерфейсу. При цьому фреймворк ReactJS поєднує швидкість та ефективність JavaScript з більш ефективним методом маніпулювання DOM для швидкого рендерингу веб-сторінок та створення високодинамічних та реактивних веб-додатків.

Завдяки здатності створювати швидкі, ефективні та масштабовані веб-додатки React завоював стабільність і популярність. Сьогодні його використовують тисячі веб-додатків, від відомих компаній до нових стартапів. Ось кілька популярних прикладів: Facebook, Instagram, Netflix, Reddit, Uber, Airbnb тощо.

Тому для моєї дипломної роботи найефективніше було використовувати фреймворк React.js, зважаючи на значні переваги у роботі поміж інших фреймворків.

2.2. Node.js.

Відповідно до вимог дипломної роботи - необхідність активного мережевого обміну даними між згенерованою системою та сервером Telegram, потреба у мікросервісах та фронтенд-додатках, а також необхідністю швидкої розробки та легкого налагодження - чат-бот був створений з використанням мови програмування JavaScript, а саме платформи Node.js.

Node.js - це одно потокова кросплатформенна бібліотека з відкритим вихідним кодом, яка використовується для запуску веб-додатків на JavaScript поза клієнтським браузером.

Ключові особливості:

1. Асинхронна модель виконання запитів

Важливою особливістю Node.js є її асинхронна природа. Асинхронність означає, що сервери, побудовані з використанням Node.js, не повинні чекати на повернення даних під час виконання різних внутрішніх запитів.

2. Неблокуючий ввід/вивід.

Node.js використовує неблокуючий ввід/вивід. Це означає, що кілька різних процесів можуть працювати паралельно, не блокуючи один одного.

3. JavaScript-рушій Google V8.

V8-це JavaScript - рушій з відкритим вихідним кодом, розроблений датським відділенням Google і розповсюджується під ліцензією BSD. Бібліотека V8 надає JavaScript-рушій Node.js, який керується Node.js через API V8 C++. V8 підтримується Google для використання з Chrome.

4. Одно потокова робота.

У Node.js всі запити збираються в циклі обробки подій і є одно поточними. Це означає, що вся програма виконується в одному потоці, починаючи з отримання запиту, виконання необхідних завдань і відправки відповіді назад клієнту. Ця особливість Node.js дозволяє уникнути

перезавантаження запитів і скоротити час обробки, що робить їх більш економічними у використанні.

5. Кросплатформенна сумісність

Node.js можна використовувати на широкому спектрі систем, від Windows до Mac OS, Linux і навіть мобільних платформ. Тому з Його допомогою можна створити самодостатнє середовище для всіх напрямків розробки.

Варто зазначити, що стандартна бібліотека Node.js наразі підтримує базовий функціонал, необхідний для виконання дипломних завдань. Прикладом є Axios-браузер на основі Promise та простий HTTP-клієнт на Node.js. Axios надає просту у використанні бібліотеку в невеликому пакеті з розширюваним інтерфейсом.

2.3. Express JS

Express - це фреймворк для веб-додатків на Node JS, який надає широкі функціональні можливості для створення веб- та мобільних додатків. Він використовується для створення односторінкових, багатосторінкових та гібридних веб-додатків.

Перевагами Express є рівень, збудований на Node.js, що допомагає оперувати маршрутами та серверами. Він допомагає заощаджувати час на кодування та все одно створювати ефективні веб- і мобільні додатки.

Особливості Express JS це:

- Швидка розробка на стороні сервера

Функції Node.js допомагають економити час.

- Проміжне програмне забезпечення

Проміжне ПЗ - це обробник запитів, який має доступ до циклу запит/відповідь програми.

- Маршрутизація

Відноситься до того, як URL-адреси кінцевої точки програми відповідають на запити клієнта.

- Шаблонування

Надає механізми, які створюють HTML шаблони на сервері, які в свою чергу створюють динамічний вміст на веб-сторінках.

- Налагодження

Express спрощує роботу тим, що визначає частину, де є помилки.

2.4. CORS

CORS або Cross-Origin Resource Sharing у Node.js - це система, що складається з передачі HTTP-заголовків, яка визначає, чи блокують браузери зовнішній код JavaScript для доступу до відповідей на запити між джерелами. Політика безпеки того самого джерела забороняє доступ до ресурсів із різних джерел.

Зазвичай передня частина програми може здійснювати виклики API лише до задньої частини у своєму джерелі. Це відоме як політика того самого походження і є досить важливим з точки зору безпеки. Усі запити, зроблені з інтерфейсу іншого походження або до серверу іншого джерела, блокуватимуться браузером. CORS дозволяє нам обійти цю політику у випадку сценаріїв, коли доступ до сторонніх ресурсів стає необхідним.

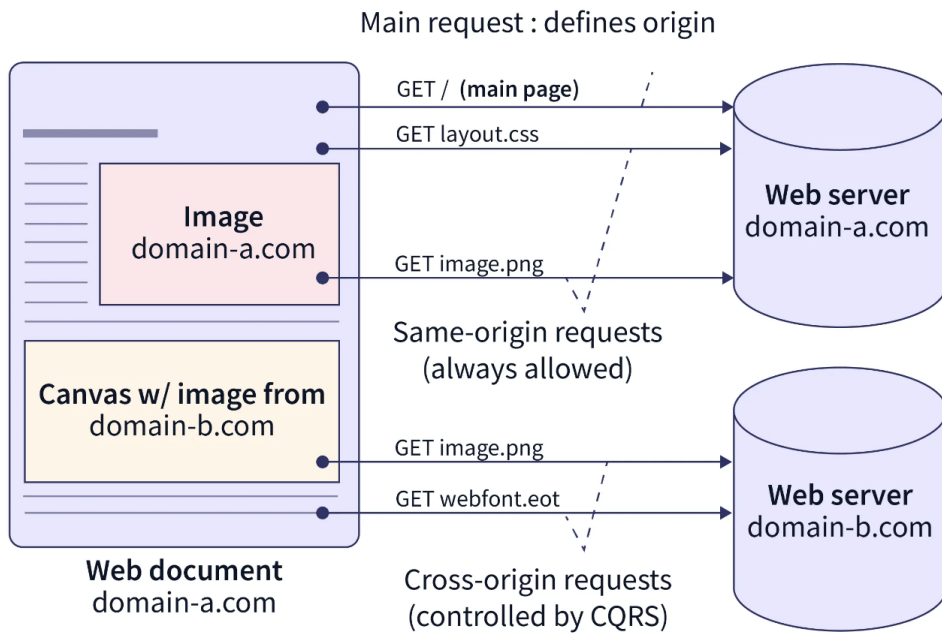


Рис. 2 Робота CORS

Браузер реалізує політику того самого джерела, яка дозволяє отримувати ресурси з його джерела (URL) і блокує ресурси із зовнішньої URL-адреси. Це можна змінити через Cors у node.js. Коли браузер робить запит, у повідомленні запиту можна вказати заголовок Origin. Якщо запит надходить до сервера з того самого джерела, тоді запит дозволено браузером і він не блокується. Однак, якщо запит надсилається на сервер за межами джерела, це називається запитом із перехресним джерелом. Під час надсилання відповіді сервер додасть до відповіді заголовок Access-Control-Allow-Origin. Значення цього заголовка має збігатися зі значенням заголовка джерела в запиті. Якщо це збіг, тоді запит дозволено. Якщо він не збігається, браузер заблокує дані відповіді. Тут користувач отримає сумнозвісну помилку CORS у консолі браузера.

Перш ніж заглиблюватися в Cors у Node.js, давайте спочатку розберемося з Origin. Як ми обговорювали раніше, зміна джерела

блокуватиме всі запити, якщо ми не реалізували CORS. Отже, що вважається зміною походження?

Origin – це заголовок запиту, який визначає URL-адресу клієнтської групи, яка зробила запит. Він має три частини:

1. Схеми напр. HTTP, HTTPS
2. Домен, напр. локальний хост, `www.corsinnodejs.com`
3. Порт напр. 8080, 8888

`http(s)://localhost:3000/`



Будь-яка зміна в наведених вище трьох частинах призведе до іншої вихідної URL-адреси та може призвести до помилки `cors`. HTTP – це не те саме, що HTTPS. Нам потрібно забезпечити відповідність джерел запиту та відповіді або налаштувати CORS у вузлі `js`, щоб ресурси не блокувалися.

Тепер перейдемо до заголовків. Простіше кажучи, заголовки використовуються для надсилання додаткової інформації із запитом або відповіддю. Ця інформація потрібна браузеру або серверу для виконання різних функцій. Заголовок запиту містить додаткову інформацію про запит. Заголовок відповіді надає додаткову інформацію про відповідь.

Origin: цей заголовок вказує URL-адресу запиту та відповіді.

Access-Control-Allow-Origin : цей заголовок вказує на URL-адресу, за якою можна робити запити між джерелами.

Отже, якщо, зовнішній клієнт надіслав запит до серверної частини і серверна частина дозволила всі вихідні URL-адреси, запит був дозволений браузером, і дані були отримані клієнтом. Код статусу 200 доведе, що запит було обслужено без помилок.

РОЗДІЛ 3. ОПИС ПРОЄКТУ

3.1. Telegram-бот

Для написання telegram-бота було використано бібліотеку telegram-bot-api. Bot API — це інтерфейс на основі HTTP, створений для розробників, які зацікавлені в створенні ботів для Telegram.

Telegram використовує власний протокол шифрування MTProto. MTProto API - це API для Telegram-додатків, що зв'язуються із сервером. Ця API є повністю відкритими, тому будь-який розробник може написати свій клієнт месенджера.

Для підключення Telegram-бота та розробки необхідного функціоналу я використала фреймворк Express JS. Це фреймворк Node.js, призначений для швидкого створення крос-платформних веб-додатків з API та полегшення використання Node.js.

Однією з переваг використання Telegram-ботів є можливість отримувати push-повідомлення про дані замовника. Замість того, щоб активно перевіряти базу даних чи очікувати електронного листа з оновленнями, адмін отримує сповіщення безпосередньо у своєму Telegram-акаунті. Простий та інтуїтивно зрозумілий інтерфейс дозволяє користувачам легко користуватися ботом, навіть якщо вони не мають спеціальних навичок або досвіду використання програмного забезпечення.

Окрім зручності та швидкості, Telegram-бот також забезпечує конфіденційність та безпеку даних замовника. Всі комунікації та передача інформації здійснюються через зашифроване з'єднання, що забезпечує захист від несанкціонованого доступу. Замовник може бути впевнений у тому, що його дані залишаються конфіденційними та безпечними.

Загалом, використання Telegram-бота для сповіщення та перегляду даних замовника є простим, легким і зручним рішенням. Він дозволяє замовнику бути постійно підключеним до своїх даних, незалежно від місця

та часу. За допомогою цього бота клієнти можуть ефективно керувати своїми замовленнями та приймати актуальні, обґрунтовані рішення.

Telegram-бот буде надсилати таку інформацію після заповнення полів в розділі “Оформлення замовлення”:

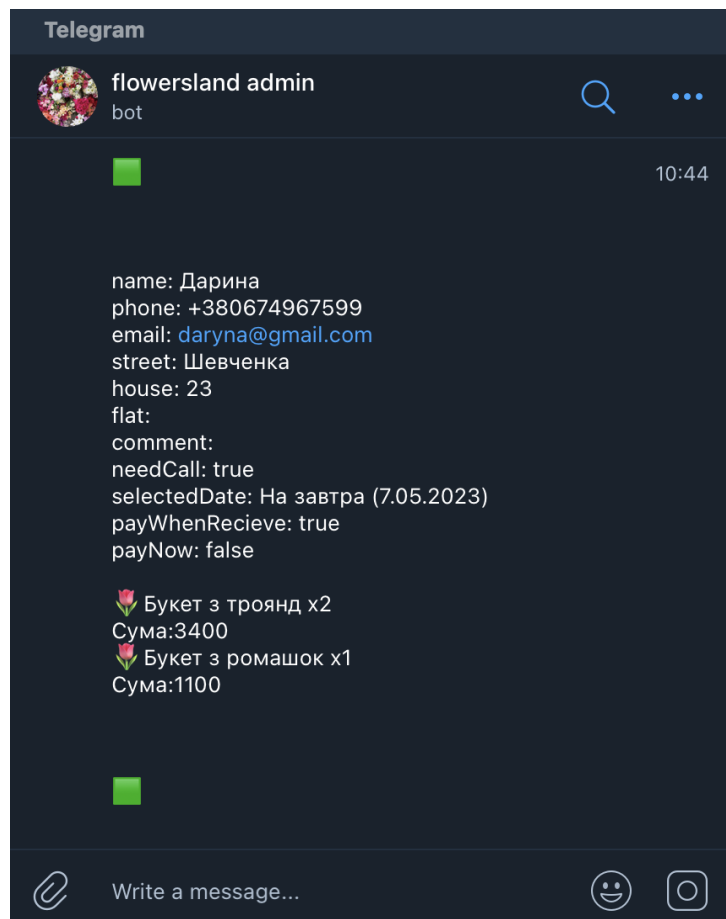


Рис. 3 Telegram-бот

Робота відбувається наступним чином:

1. Користувач взаємодіє з фронтендом (наприклад, відправляє запити, вводить команди тощо).
2. Фронтенд передає дані та команди до Bot API через бекенд.
3. Бекенд отримує запити від фронтенду і генерує відповіді для них, використовуючи функціональність бібліотеки telegram-bot-api.
4. Замість того, щоб взаємодіяти з Telegram-сервером безпосередньо, бекенд використовує вбудований функціонал бібліотеки, що дозволяє відправляти HTTP-запити на сервери Telegram.

5. Бекенд генерує та відправляє HTTP-запити на сервери Telegram, використовуючи необхідні методи та параметри, а саме axios.
6. Telegram-сервер обробляє отриманий від бота запит і повертає відповідь.
7. Бекенд отримує відповіді від серверів Telegram та обробляє їх, наприклад, для подальшої обробки або відправки назад до фронтенду через API.

Такий підхід дозволяє взаємодіяти між фронтендом і сервером Telegram за допомогою Telegram-бота, який працює в якості посередника. Таким чином, фронтенд може зв'язатися з бекендом, а бекенд відправляє запити на сервери Telegram для виконання необхідних дій.

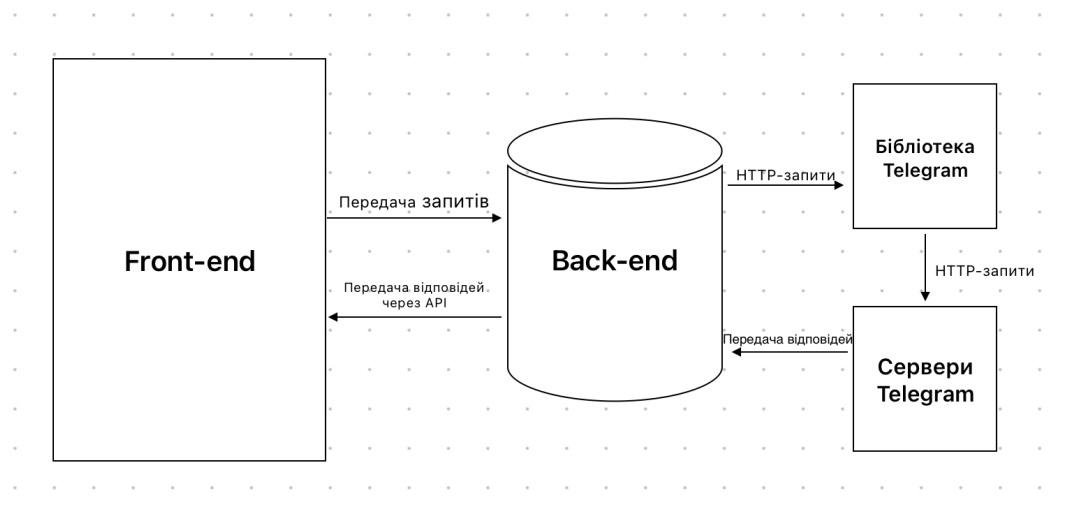


Рис. 4 Схема опрацювання даних від Front-end до Back-end

3.2. Робота з Firebase

Firebase – це платформа для розробки мобільних і веб-додатків, яка дозволяє користувачам швидко створювати високоякісні додатки, залучати лояльних користувачів і збільшувати прибутки. Платформа включає в себе низку інтегрованих модулів, включаючи мобільні сервери, аналітику, інструменти для розробки та монетизації, що дозволяють досягти максимального успіху додатків.

Firestore Database - це продукт Google Firebase. Гнучка і масштабована хмарна база даних NoSQL, Firestore дозволяє зберігати дані ваших додатків і синхронізувати їх за допомогою слухачів в режимі реального часу. Вона добре працює в будь-якій ситуації, незалежно від підключення до Інтернету.

База даних Firestore має такі функції:

Гнучкість: дані можуть зберігатися в документах, організованих у колекції. Цей підхід дозволяє мати декілька документів (в межах однієї колекції), з різною структурою.

Запити: ви можете створювати запити для отримання конкретних документів або для отримання всіх документів у колекції.

У режимі реального часу: база даних Firestore допомагає виконувати синхронізацію даних, коли дані оновлюються на підключених пристроях.

Офлайн: запис у базу даних навіть у режимі офлайн. Коли програма не підключена до Інтернету, дані зберігаються на локальному пристрої, а коли пристрій повертається в режим онлайн, дані синхронізуються з хмарною базою даних Firestore.

Масштабування: база даних Firestore працює в інфраструктурі Google.

У своєму дипломному проєкті я використовую базу даних для зберігання товарів. У таблиці я тримаю такі значення: назва, опис, категорія, ціна, зображення та кількість, яка початково завжди дорівнює нулю.

Таблиця товари

Document ID	Ідентифікатор товару (string)
title	Назва товару (string)

desc	Опис товару (string)
category	Категорія товару (string)
img	Адреса посилання зображення (string)
price	Ціна товару (number)
count	Кількість (number)

У firebase вона матиме такий вигляд:

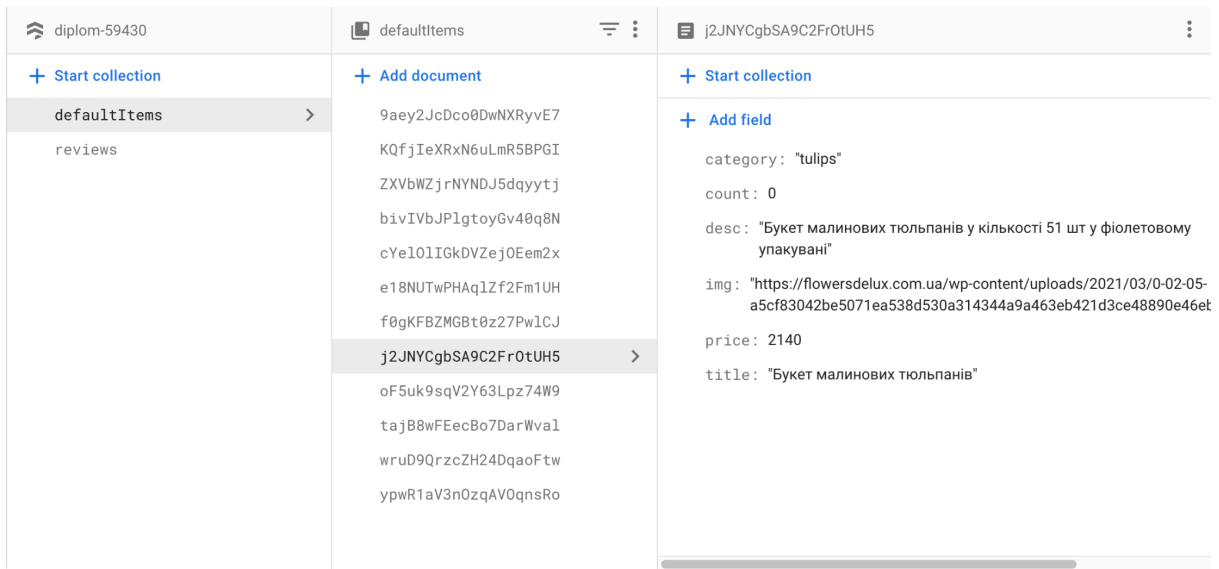


Рис. 5 Колекція “каталог” у Firestore

Я створюю колекцію, яка має назву defaultItems, в неї додаю потрібні мені поля та значення.

Add a document

Parent path
/defaultItems

Field	Type	Value
title	string	Букет тюльпанів
desc	string	Букет з рожевих
category	string	tulips
img	string	https://flowersde
price	string	3498
count	string	0

Cancel Save

Рис. 6 Додавання товарів у таблицю

Document ID генерується автоматично випадковим чином. Також є можливість редагувати значення поля. Тому це дуже зручна та проста у використанні база даних.

Для виводу даних на сторінку, я ініціалізую firebase у коді та використовую потрібні SDK для продуктів, які використовую. Для цього firebase генерує унікальні ключі, які забезпечують повну приватність та надійність.


```
src > config > JS firebase.js > ...
1  import { initializeApp } from "firebase/app";
2  import { getAuth, GoogleAuthProvider } from "firebase/auth";
3  import { getFirestore } from "firebase/firestore";
4
5  const firebaseConfig = {
6    apiKey: "AIzaSyB...gR7-J1KGQng",
7    authDomain: "dip...firebaseapp.com",
8    projectId: "d...80",
9    storageBucket: "d...appspot.com",
10   messagingSenderId: "...0363",
11   appId: "1...a550fe5634026d6f2"
12 };
13
14 const app = initializeApp(firebaseConfig);
15 export const auth = getAuth(app);
16 export const googleProvider = new GoogleAuthProvider();
17
18 export const db = getFirestore(app);
19
```

Рис. 7 Ініціалізація firebase у кодї

Щоб отримати ці дані я використовую хук useEffect.

```
27 const defaultItemsCollectionRef = collection(db, "defaultItems");
28
29 useEffect(() => {
30   const getFlowerList = async () => {
31     try {
32       const data = await getDocs(defaultItemsCollectionRef);
33       const filteredData = data.docs.map((doc) => ({
34         ...doc.data(),
35         id: doc.id,
36       }));
37       setAllItems(filteredData);
38       setDefaultItems(filteredData);
39     }
40     catch (err) {
41       console.error(err);
42     }
43   };
44
45   getFlowerList();
46 }, []);
```

Рис. 8 Приклад роботи useEffect

Наступна таблиця називається reviews, вона містить такі дані:

Document ID	Ідентифікатор відгуку (string)
User ID	Ідентифікатор користувача (string)
content	Вміст відгуку (string)
user	Електронна пошта користувача (string)
createdAt	(date)

Ця таблиця зв'язана за допомогою User ID з Firebase Auth. Це служба, яка аутентифікує користувачів. Вона використовує тільки код, який знаходиться на стороні клієнта. Firebase Auth також включає в себе систему управління користувачами, що в свою чергу допомагає розробникам увімкнути автентифікацію користувача за допомогою входу з електронної пошти та пароля, що зберігаються в Firebase. Ця умовна таблиця має такий вигляд:

Identifier	Providers	Created ↓	Signed In	User UID
dmytro_099@ukr.net	✉	May 29, 2023	May 29, 2023	laLJeLsANZaRDho9MWiFUX8jmi92
user14@gmail.com	✉	May 29, 2023	May 29, 2023	mIM0jYR94hcdFV0rSjB4hxHonCb2
user1@gmail.com	✉	May 29, 2023	May 29, 2023	WffG09IIYeUGybWEjVEDG8yteHG2
kkfkffk@kfkfkf.ggj	✉	May 28, 2023	May 28, 2023	EDGNNEIIUiZa5WwR2qI7riPothE2
usergkgkgk@gmail.com	✉	May 28, 2023	May 28, 2023	T25iyV0iMRcxFY69CFThRJ6s2QS2

Рис. 9 Служба Firebase Auth

Спочатку для отримання даних я використовую хук useEffect, так як на рис 7, а потім використовую оператор await, який змушує функцію, оголошену з використанням оператора async, чекати на виконання Promise і продовжувати виконання після повернення Promise значення. Таким чином дані вписані через сайт ми отримуємо у таблиці бази даних.

```

const onSubmitReview = async () => {
  try {
    await createUserWithEmailAndPassword(auth, email, '12345678');
    await addDoc(reviewsCollectionRef, {
      content: newReview,
      user: auth.currentUser.email,
      createdAt: serverTimestamp(),
      userId: auth?.currentUser?.uid,
    });
    getReview();
  } catch (err) {
    console.error(err);
  }
};

```

Рис. 10 Показ передачі даних з Front-end на Firestore

Так виглядає таблиця, після отримання даних.

Document ID	content	createdAt	user	userId
DcZi4UJOZHf2ixvkFUy4	"Дякую за швидку доставку :)"	May 29, 2023 at 12:27:13 AM UTC+3	"user14@gmail.com"	"miMOjYR94hcdFv0rSjB4hxHonCb2"
JDDc8854uyXFH0zwWEfL	"Консультанти магазину допомогли обрати найкращий букет для свята."	May 29, 2023 at 12:28:49 AM UTC+3	"dmytro_099@ukr.net"	"laLJeLsANZaRdho9MWiFUX8jmi92"
U2CхwTbdizquNEHg63xk	"Гарний асортимент!"	May 29, 2023 at 12:26:50 AM UTC+3	"user1@gmail.com"	"WFFGO9IIYeUGybWEJVEDG8yteHG2"
zwzYmkSsE9DvCRulo1Bq	"soooooooooo"	May 28, 2023 at 8:30:35 PM UTC+3	"usergkgkgk@gmail.com"	"T25iyV0iMRcxFY69CFTThRJ6s2QS2"

Рис. 11 Таблиця з даними

3.3. UI частина інтернет-магазину

Для UI частини веб-сайту я використовувала Material-UI. Material-UI - це популярна бібліотека інтерфейсу користувача для React-програм, яка дозволяє розробникам швидко створювати естетичні інтерфейси з використанням компонентів, заснованих на дизайні Material Design від Google.

Material-UI пропонує безліч готових компонентів, таких як кнопки, форми, картки, модальні вікна та навігаційні панелі. Ці компоненти мають чистий сучасний дизайн, який відповідає принципам Material-Design.

Бібліотека Material-UI також пропонує інші корисні функції, такі як адаптивний дизайн, теми оформлення, інтеграція зі станами програми та підтримка анімації. Вона добре задокументована і має активну спільноту розробників, яка може допомогти з питаннями та вирішенням проблем.

Material-UI робить розробку веб-інтерфейсів швидшою та простішою, допомагаючи створювати красиві та функціональні додатки. Він використовується в багатьох проєктах і має велику кількість прихильників у спільноті розробників React.

Всі використовувані іконки взяті з Font Awesome, набору шрифтів та іконок на основі CSS та Less. Станом на 2020 рік 38% сайтів, що використовують сторонні шрифтові скрипти, використовують Font Awesome, поступаючись за популярністю лише Google Fonts.

3.4. Функціонал інтернет-магазину

Спроектований Інтернет-ресурс містить та оперує інформацією про флористичний магазин, який називається “FlowersLand”. Магазин пропонує великий асортимент квітів. Усі позиції завжди свіжі та цікаві. Це все представлено у вигляді каталогу та розділено по категоріям, тобто за назвою квітки. Кожен тип категорії зберігається в базі даних, має свій набір властивостей та параметрів (назва товару, опис, фото та ціну).

Розпочнемо з головної сторінки.

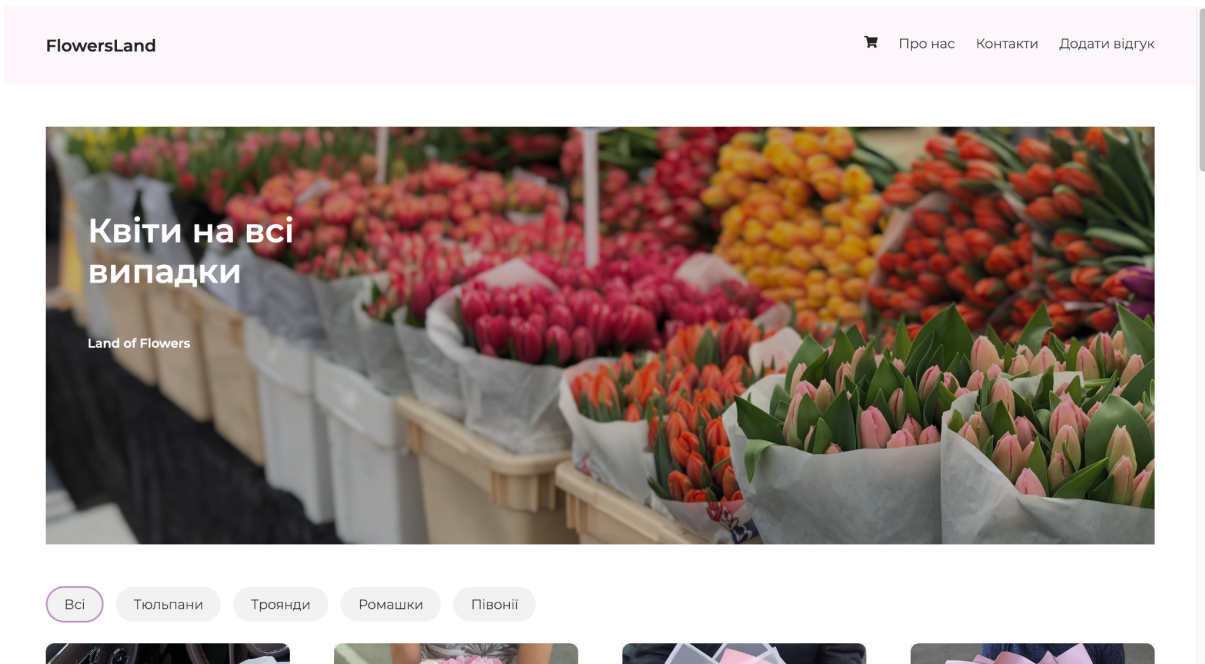


Рис. 12 Головна сторінка

Хедер надає можливість переходити між сторінками. Наприклад, сторінка “Про нас”. Ця сторінка містить опис магазину та карусель фото.

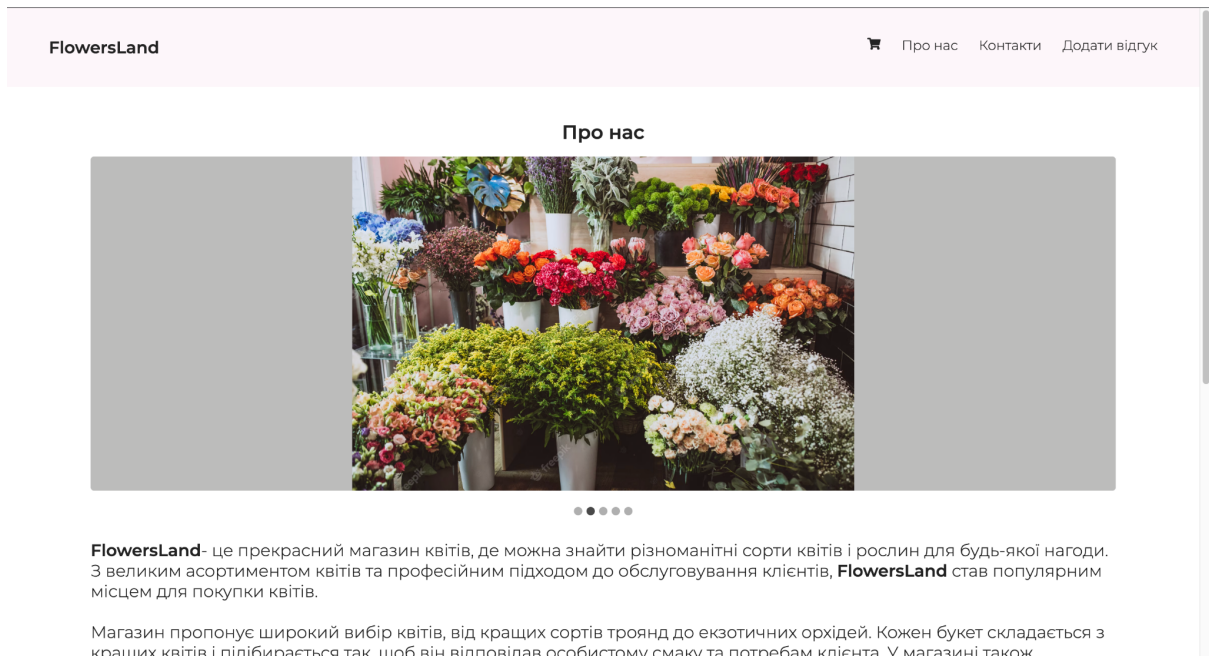


Рис. 13 Сторінка “Про нас”

Сторінка “Контакти” містить контактні дані та зображення карти, проте після натиску на зображення відкривається посилання на Google карти.

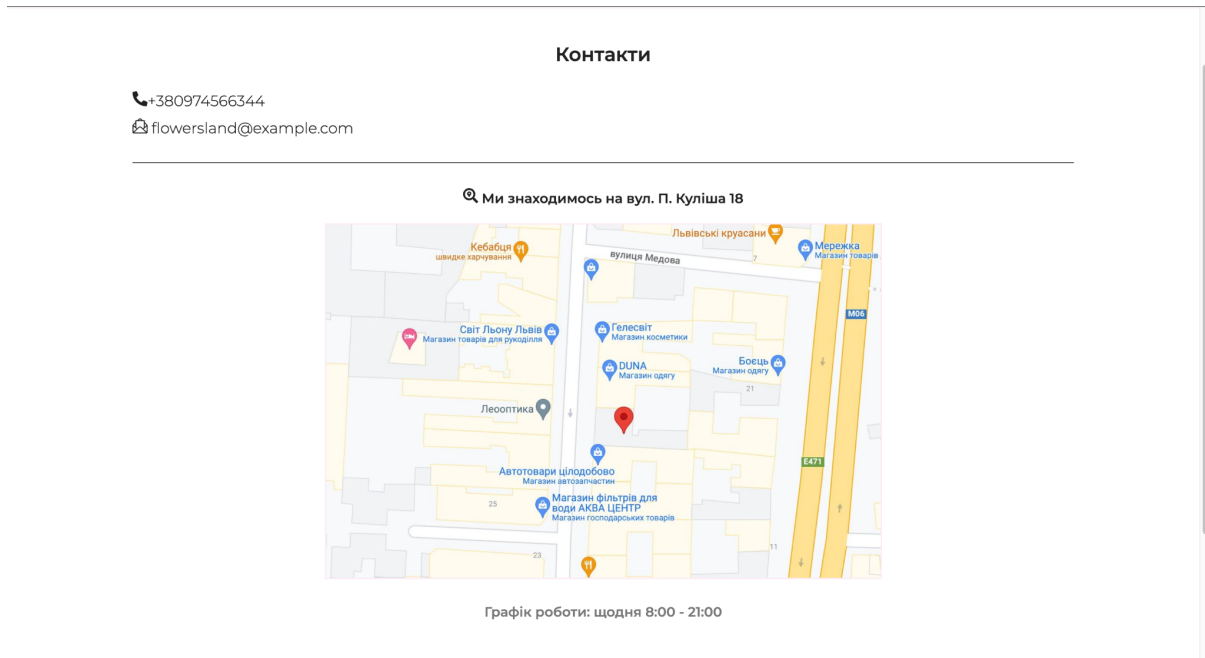


Рис. 14 Сторінка “Контакти”

Сторінка “Додати відгук” дає можливість користувачеві переглянути чужі відгуки або ж залишити свій після введення електронної пошти. Дані з цієї сторінки зберігаються у базі даних.

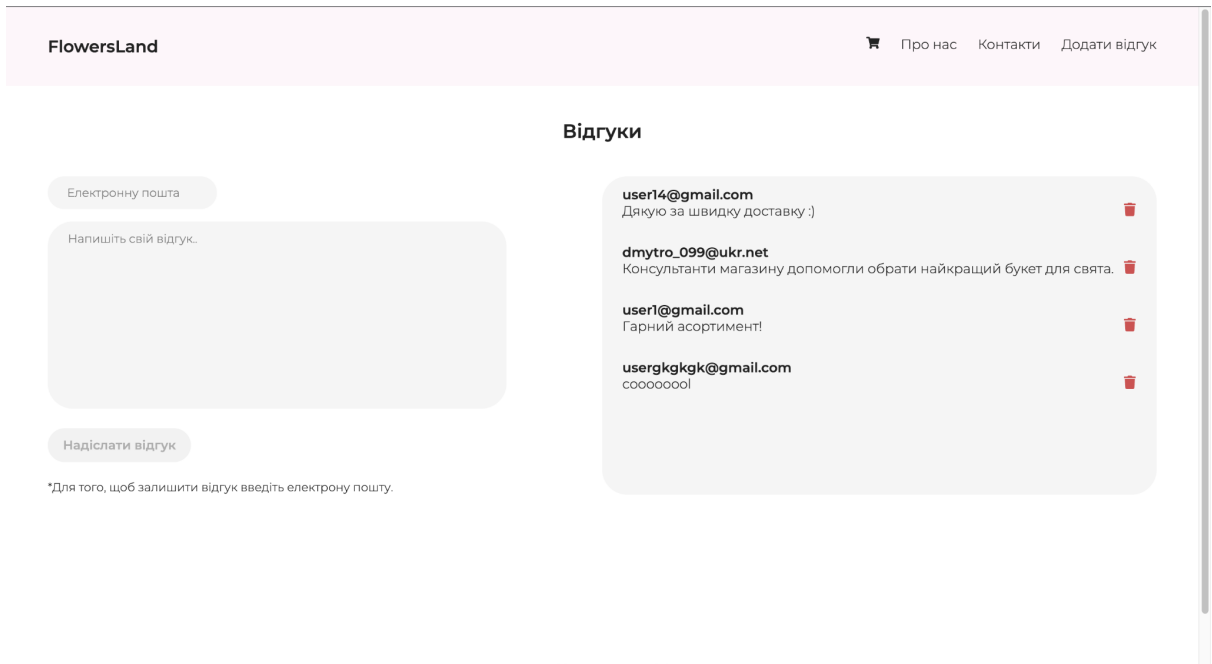


Рис. 15 Сторінка “Відгуки”

На головній сторінці містяться товари, які можна поділити за категоріями. Сортування товарів відбувається нових до старих.

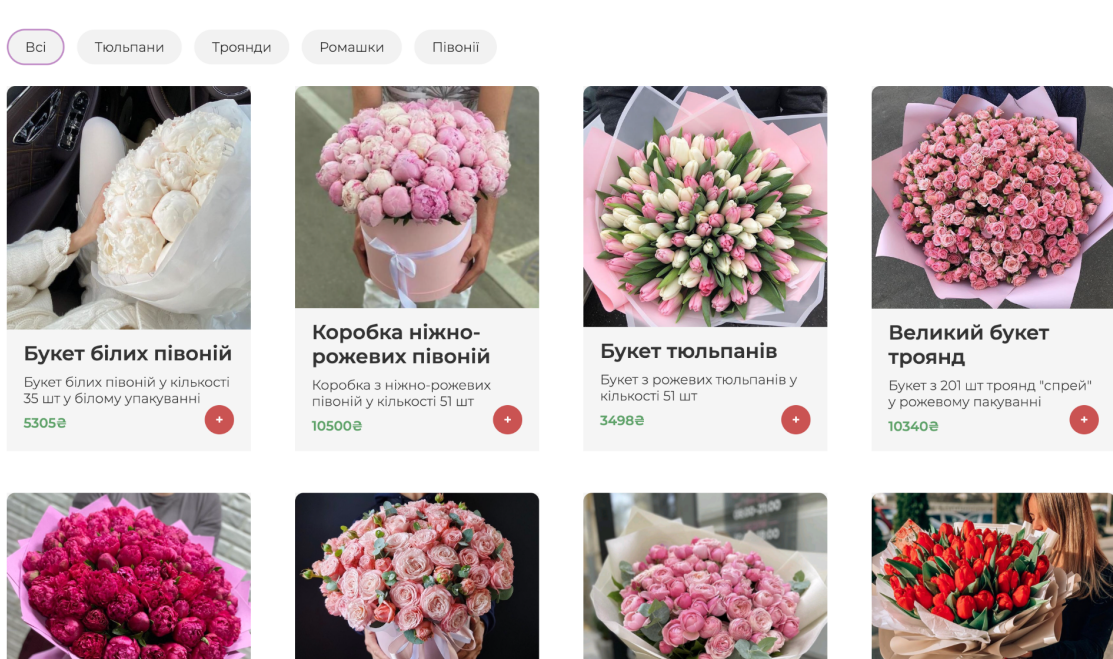


Рис. 16 Каталог товарів

При натиску на зображення товару ми відкриваємо його у великому вікні.

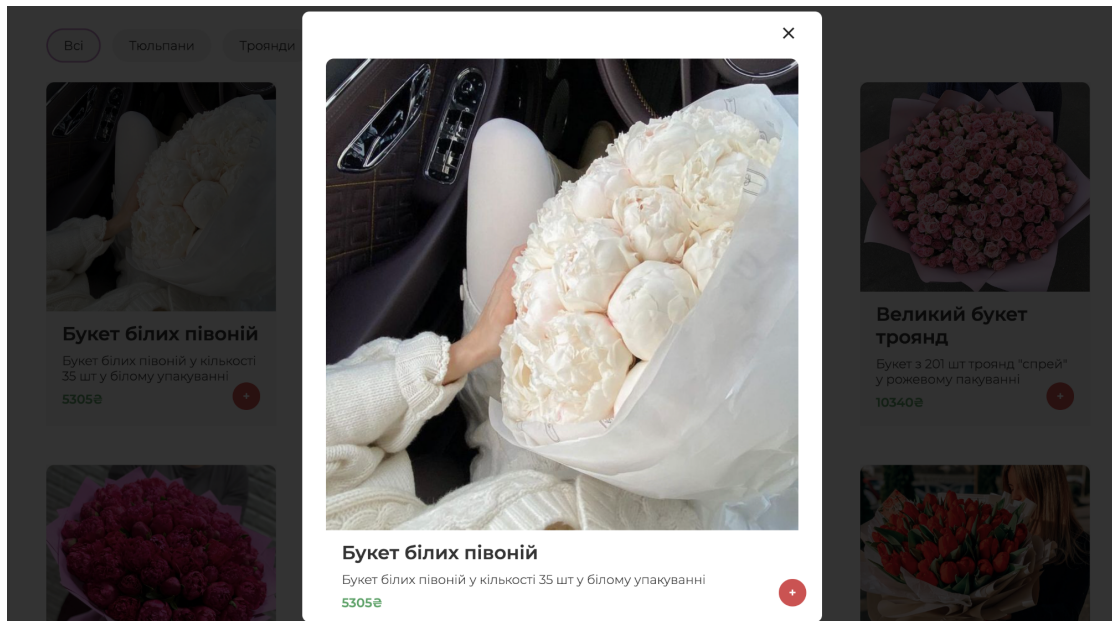


Рис. 17 Вигляд товару у великому вікні

Після додавання товару в кошик, ми можемо його переглянути після натиску на іконку корзини. Ми можемо змінити кількість, видалити або замовити.

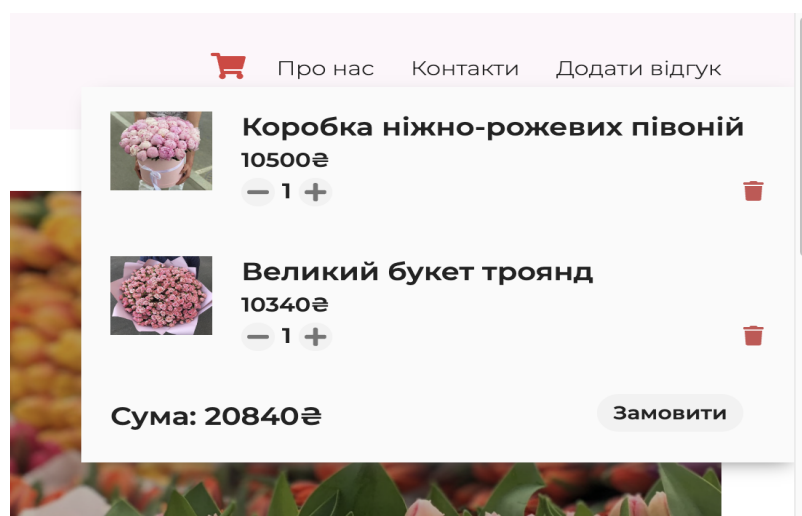


Рис. 18 Вигляд корзини в хедері

Ми можемо змінити кількість, видалити або замовити. Кнопка “Замовити” перекине нас на сторінку “Ваша корзина”, де буде на весь екран видно наші товари, та дасть нам ще можливість змінити кількість або видалити.

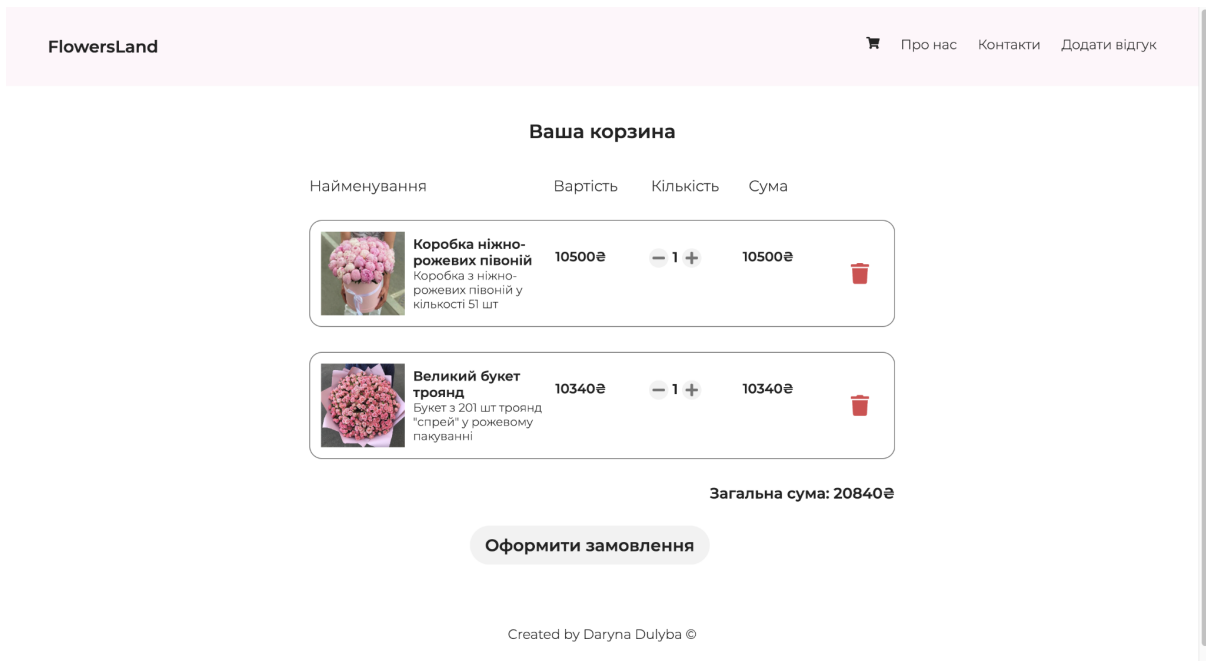


Рис. 19 Сторінка “Ваша корзина”

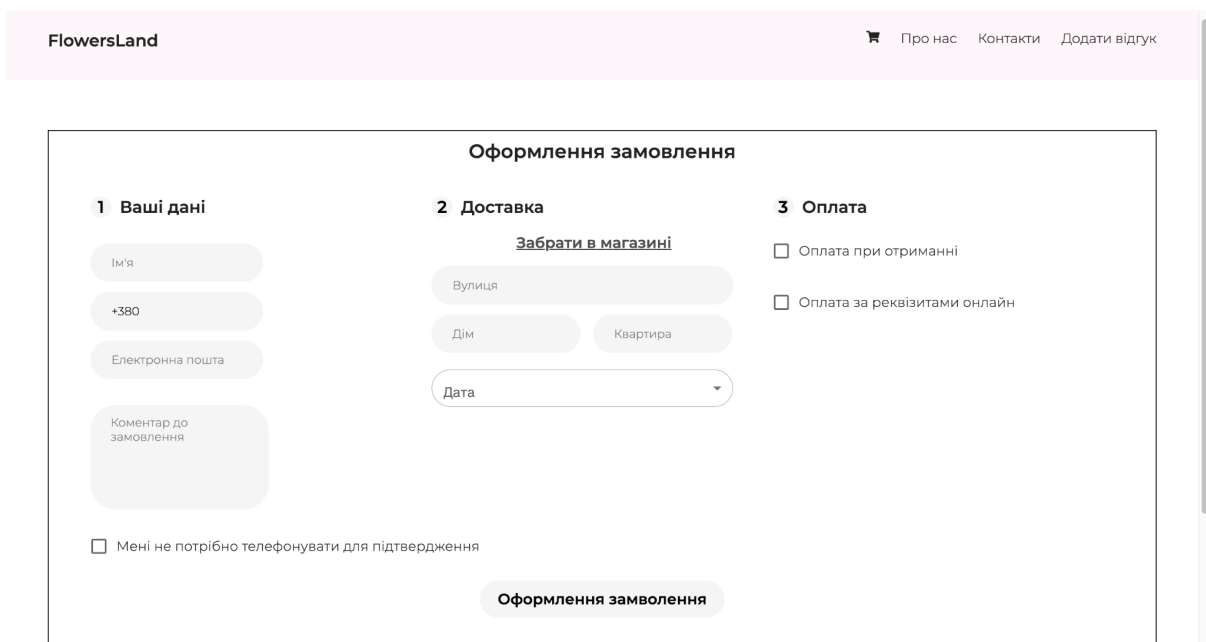


Рис. 20 Сторінка “Оформлення замовлення”

Оформлення замовлення

1 Ваші дані

Дарина

+380961234567

user_12@gmail.com

Коментар до замовлення

Мені не потрібно телефонувати для підтвердження

2 Доставка

Доставка кур'єром

📍 FlowersLand (м. Львів, вул. П.Куліша 18)

Дата: 8.05.2023

*Для замовлення в іншу дату прошу за телефонувати за номером, що вказаний в розділі "Контакти"

3 Оплата

Оплата при отриманні

Натисніть для переходу на сторінку реквізитів

Оформлення замовлення

Рис. 21 Сторінка “Оформлення замовлення” із заповненими полями

Якщо ми вибираємо пункт “Оплата за реквізитами онлайн”, ми натискаємо та переходимо на сторінку ПриватБанку, де можна здійснити оплату. Кнопка “Оформлення замовлення” відправляє наші дані в telegram-бот, і я отримаю таку інформацію.

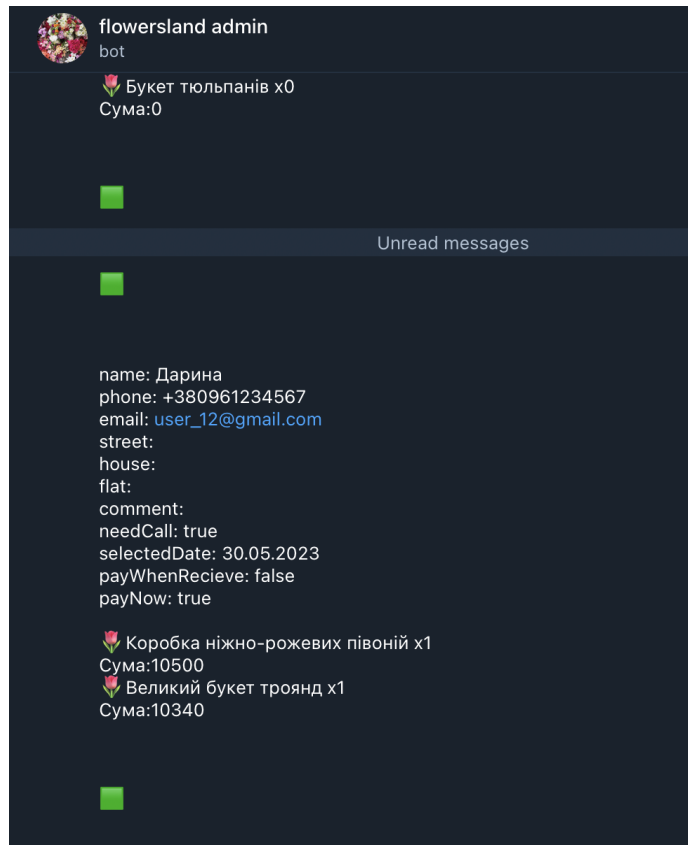


Рис. 22 Сповіщення в telegram-боті після оформлення замовлення

РОЗДІЛ 4. ХОСТИНГ ІНТЕРНЕТ-МАГАЗИНУ

Для хостингу інтернет-магазину я використовую Firebase Hosting. Firebase Hosting - це статичний і динамічний веб-хостинг, запущений 13 травня 2014 року для розміщення статичних файлів, таких як CSS, HTML і JavaScript, а також динамічної підтримки Cloud Functions і Node.js. Сервіс також підтримує динамічну підтримку Node.js за допомогою Cloud Functions. Сервіс передає файли через мережу доставки контенту (CDN) за допомогою HTTPS і SSL-шифрування; Firebase підтримує Fastly, CDN, і забезпечує підтримку CDN для хостингу Firebase.

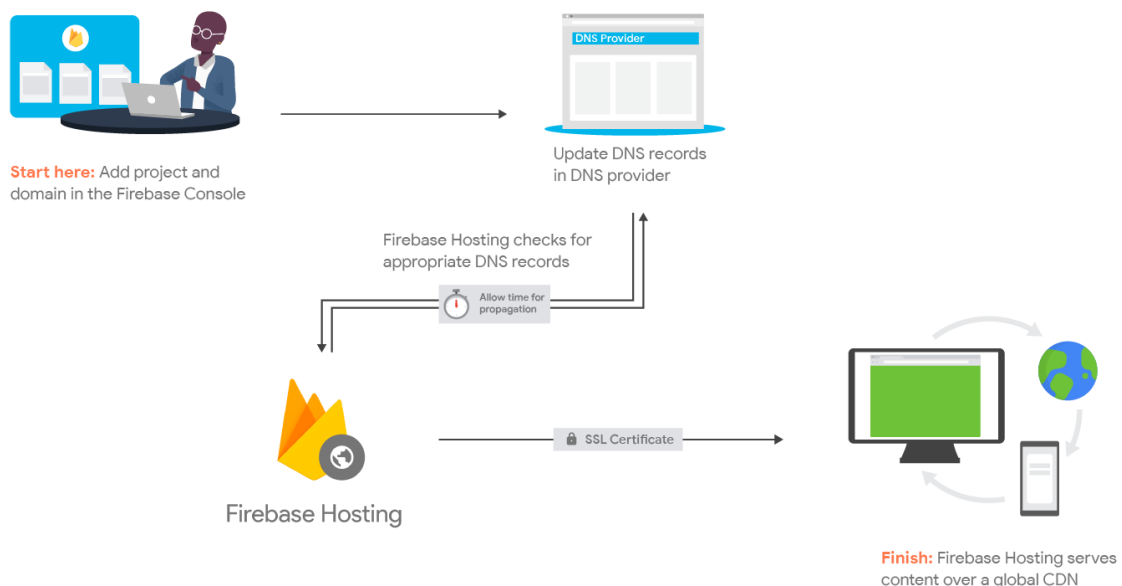


Рис. 23 Схема роботи Firebase Hosting

ВИСНОВОК

В рамках дипломної роботи був розроблений інтернет-магазин, який надає зручну можливість покупки квітів без потреби виходити з дому. Сайт дозволяє покупцям переглядати зображення, назву, ціну та опис товару, а також оформляти замовлення, залишати відгуки тощо.

Усі дані про товари зберігаються в базі даних Firestore, що є частиною платформи Firebase. Адміністратор магазину має повний контроль над товарами, оскільки може додавати, редагувати та видаляти їх безпосередньо через Firebase Firestore.

Одна зі сторінок магазину дозволяє клієнтам залишати відгуки. Для цього користувачі повинні ввести свою електронну пошту. Ця функція пов'язана з Firebase Auth, що забезпечує аутентифікацію введеної пошти і зберігання її в хмарному сховищі. Всі відгуки, які були введені, також зберігаються в базі даних. На сайті є можливість видаляти відгуки, а база даних Firestore містить інформацію про користувача, який залишив відгук, дату, час і сам відгук.

Основою для роботи інтернет-магазину стала платформа Firebase, зокрема база даних Firestore та служба Auth. Це дозволило зберігати всю необхідну інформацію про товари, відгуки та користувачів в хмарному сховищі та забезпечувати безпеку та аутентифікацію користувачів.

Окремо варто відзначити зручний та привабливий дизайн сайту, розроблений самостійно. Використання бібліотек React Icons та Material-UI додало до сайту естетичного вигляду та покращило його користувацький інтерфейс.

Під час роботи над дипломною роботою я продемонструвала високий рівень володіння технологіями, такими як React JS, Node.js та Firebase. Успішно використала їх для реалізації потрібного функціоналу, а також вдосконалила свої навички у використанні цих інструментів.

У результаті, інтернет-магазин виявився функціональним, зручним у використанні та привабливим для користувачів. Розробка цього проекту дозволила мені отримати цінний досвід та поглибити знання у сфері веб-розробки з використанням сучасних технологій.

Готовий код можна переглянути за посиланням <https://github.com/kpm-lnu/student-applications/commit/7372e2f38fe7a18d5c6d70f880d733fe88f3cb06>.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “DIGITAL 2023: GLOBAL OVERVIEW REPORT” [Онлайн].
Доступно:
<https://datareportal.com/reports/digital-2023-global-overview-report>
(Дата відвідування: 01.11.2022)
2. Simplilearn. “What is Express JS in Node JS?” [Онлайн]. Доступно:
<https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js>
(Дата відвідування: 20.02.2023)
3. Blog. “What is React.js?” [Онлайн].
Доступно: <https://blog.hubspot.com/website/react-js>
(Дата відвідування: 06.04.2023)
4. Dan-IT. “Що таке Node JS простими словами.” [Онлайн]. Доступно:
<https://dan-it.com.ua/uk/blog/chto-jeto-takoe-node-js-prostymi-slovami/>
(Дата відвідування: 15.01.2023)
5. Code Capsules. “Create and Host a Telegram Bot with Node.js.” [Онлайн]. Доступно:
<https://codecapsules.io/docs/tutorials/create-nodejs-telegram-bot/>
(Дата відвідування: 21.03.2023)
6. Scaler. “CORS in Node.js.” [Онлайн].
Доступно: <https://www.scaler.com/topics/nodejs/cors-in-node-js/>
(Дата відвідування: 21.03.2023)
7. Flutterflow. “Firestore Database.” [Онлайн]. Доступно:
<https://docs.flutterflow.io/data-and-backend/firebase/firestore-database>
(Дата відвідування: 04.05.2023)
8. Wikipedia. “Firebase.” [Онлайн]. Доступно:
<https://uk.wikipedia.org/wiki/Firebase>
(Дата відвідування: 04.05.2023)

